Titolo Volume
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000001
pag. 15–?? (xx 9999)

A Review of Mathematical and Computational Aspects of CSIDH-based Algorithms

Luciano Maino, Federico Pintore

1. Introduction

Modern cryptography requires the security of every cryptosystem to be formally proven. The security of public-key primitives necessitates assuming that some mathematical problems cannot be solved in polynomial time. Many of the public-key cryptosystems currently in use rely on the (presumed) hardness of factoring integers or computing discrete logarithms in finite cyclic groups. Even though the hardness of these problems was considered trustworthy for a long time, in 1994 Shor [37] proposed Las Vegas polynomial-time algorithms for factoring integers and computing discrete logarithms on quantum computers - machines that exploit quantum mechanical phenomena to store data and perform computations.

The growing research on quantum computing has made concrete the prospect of building, in the near future, quantum computers able to implement Shor's algorithm and, consequenlty, to make most of the public-key schemes currently deployed totally unsafe. To respond to this threat, new mathematical problems (supposed to be) hard to solve even for quantum computers have been introduced. The line of cryptographic research, named post-quantum cryptography, which works with these class of problems to design quantum-safe cryptosystems has seen an exponential growth in recent years.

The (supposedly) quantum-resistant cryptosystems that have been proposed so far could be merged into five large families: lattice-based, code-based, hash-based, multivariate and isogeny-based schemes. In this work, we will restrict our attention to the latter family, which is derived from classical number-theoretical results and, despite being the most recent, is particularly appealing as it enjoys short keys.

The first isogeny-based cryptosystem to be proposed was a non-interactive key-exchange protocol detailed by Couveignes in a talk given in 1997 [13]. Until 2006 this work only spread privately, and the system was re-discovered independently by Stolbunov and Rostovtsev [35]. The scheme exploits the free and transitive action of the ideal class group $\mathcal{C}\ell(\mathcal{O})$ on the $\mathbb{F}_{p^m}$-isomorphism classes of ordinary elliptic curves over $\mathbb{F}_{p^m}$ having endomorphism ring isomorphic to an order $\mathcal{O}$ (of an imaginary quadratic field) and a given trace of Frobenius. Despite its theoretical interest, a subexponential quantum attack - designed by Childs *et al.* [11] using the commutativity of $\mathcal{C}\ell(\mathcal{O})$ - together with the inefficiency of the scheme locate it out of the realm of practicality.

To overcome the efficiency issues, in 2018 Castryck *et al.* [8] reshaped the idea of Couveignes, Stolbunov and Rostovtsev using supersingular elliptic curves rather than ordinary ones. Indeed, given a prime p and a supersingular elliptic curve E over $\mathbb{F}_p$, the ring $\mathsf{End}_{\mathbb{F}_p}(E)$ composed by the endomorphisms of E defined over $\mathbb{F}_p$ is still an order $\mathcal{O}$ in a quadratic field (specifically, $\mathbb{Q}(\sqrt{-p})$). Furthermore, the ideal class group $\mathcal{C}\ell(\mathcal{O})$ acts freely and transitively on the set of ($\mathbb{F}_p$-isomorphism classes of) all supersingular elliptic curves E over $\mathbb{F}_p$ for which $\mathsf{End}_{\mathbb{F}_p}(E)$ is isomorphic to $\mathcal{O}$. The resulting protocol is named CSIDH, which stands for Commutative Supersingular Isogeny Diffie-Hellman and is pronounced "sea-side". CSIDH enjoys a faster key exchange than the original scheme, since the structure of the rational groups of supersingular elliptic curves over $\mathbb{F}_p$ allows to efficiently compute the action of a small set of class group elements.

To be more precise, computing the action of an arbitrary element $[\mathfrak{a}] \in \mathcal{C}\ell(\mathcal{O})$ has exponential complexity. However, the set of rational points $E(\mathbb{F}_p)$ of a supersingular elliptic curve E over $\mathbb{F}_p$ has cardinality equal to $p+1$ and, considering a prime $p = 4\ell_1\ell_2\cdots\ell_n - 1$ with $\ell_1, \ell_2, \ldots, \ell_n$ small odd primes, a special element $[\mathfrak{I}_i] \in \mathcal{C}\ell(\mathcal{O})$ can be associated to each prime ℓ_i. The action of these elements (and their inverses) can be computed very efficiently, since it is determined by an isogeny whose kernel is the unique subgroup of $E(\mathbb{F}_p)$ of order ℓ_i. Here, we recall that an isogeny is a rational map between two elliptic curves which is also a homomorphism of groups. As a consequence, it is possible to work on the base field $\mathbb{F}_p$ to efficiently compute the action of the elements in $\mathcal{C}\ell(\mathcal{O})$ of the form $\prod_{i=1}^{n}[\mathfrak{I}_i]^{e_i}$ - where the integral exponents e_i are chosen from some small interval $[-B, B]$ - in a sequential way.

The CSIDH scheme restricts itself to these elements. Indeed, the private key of a user is a vector $(e_1, \ldots, e_n) \in [-B, B]^n$, and a key exchange requires the computation of the action of $[\mathfrak{a}] = \prod_{i=1}^{n} [\mathfrak{I}_i]^{e_i}$. This computation consists in calculating a chain of "atomic" isogenies (one for each occurrence of the factors $[\mathfrak{I}_i]$), and it is the most expensive step of the protocol. Recently, several strategies have been proposed to speed-up this computation [3, 7, 28, 31, 33] and to make its runnning time independent of the private key in order to avoid active attacks [1, 5, 9, 10, 21, 27, 32].

1.1. Our contribution

The above mentioned advancements appeared in several distinct papers, each one using its own terminology and building on one or more of the previous contributions. This research line has consequently evolved into multiple sub-branches, making it laborious to keep track of all the developments. The aim of this paper is to provide a resource that summarises the most relevant of the recent contributions in a unified treatment. In doing this, we divide the presentation into three main themes: speed-up of the class group computation, constant-time class group computation and computations in a class group with known structure. In the following, we outline the results we will discuss for each of the themes.

Speed-up of the class group computation

In the original CSIDH implementation, for each atomic isogeny φ corresponding to the action of some $[\mathfrak{I}_i]$ it is necessary to determine a rational point P, lying on the curve from which the isogeny is originating, of order divisible by ℓ_i. Usually the order of P is also divisible by some of the others odd primes $\ell_1, \ldots, \ell_{i-1}, \ell_{i+1}, \ldots, \ell_n$, meaning that $\varphi(P)$ could still be used to compute some other atomic isogenies. Using $\varphi(P)$ is cheaper than determining a new rational point, therefore improving its computation is of great interest.

In [28] Meyer and Reith proposed a trick to reduce the number of field operations for computing $\varphi(P)$. Furthermore, they highlighted how the birational equivalence between Montgomery curves and Twisted Edwards curves can improve the calculation of the image curve of φ.

In [33] Onuki and Takagi showed that $\prod_{i=1}^{n}[\mathfrak{I}_i]^3 = [(1)]$ and derived an efficient procedure to compute the action of $\prod_{i=1}^{n}[\mathfrak{I}_i]$. This can be exploited to accelerate the execution of the actions corresponding to vector $(e_1, \ldots, e_n)$ having Hamming weight n.

Nakagawa *et al.* [31] proposed to use an L_1-norm ball as secret-key space to obtain a secure variant of CSIDH whose average execution cost is close to the optimal one.

Finally, Castryck and Decru [7] used a different approach to speed up the class group computation: they transposed the CSIDH scheme to a set of supersingular elliptic curves different from that used in CSIDH. The resulting scheme for a bespoke prime p is %5.68 faster than the CSIDH protocol for a prime of a similar size.

Constant-time class group computation

In the implementation of the CSIDH protocol proposed in [8], the running time to compute the action of $\prod_{i=1}^{n}[\mathfrak{I}_i]^{e_i}$ heavily depends on the private key $(e_1, \ldots, e_n)$. Consequently, an active attacker having access to timing data could leak information about the private key.

To tackle this issue, Bernstein *et al.* presented in [5] a constant-time implementation that, for each private key, executes the same number of field operations, with a negligible failure probability. Building on some of the techniques introduced in [5], Meyer *et al.* [27] devised a no-failure implementation whose number of field operations is independent of the private key but may vary due to the randomness used. In particular, in their implementation a fixed number of isogenies is computed for each of the small odd primes $\ell_1, \ldots, \ell_n$, and a key space lying in $\mathbb{N}^n$ is considered.

The implementation proposed by Onuki *et al.* in [32] retains some of the techniques from [27] while allowing secret keys to have negative entries. In particular, for the computation of each atomic isogeny, two points - one on the curve from where the isogeny is emanating and the other on an *associate curve* - are used.

Hutchinson *et al.* [21] and Chi-Domínguez and Rodríguez-Henríquez [10], to further speed up the constant time computation, independently extended to the CSIDH setting the optimal strategies introduced in [22].

Computations in a class group with known structure

The similarities of CSIDH with the standard Diffie-Hellman protocol have led researchers to use it as a bulding block for new isogeny-based cryptosystems and, in particular, digital signatures.

Stolbunov [39] was the first to propose an isogeny-based digital signature scheme, but his proposal requires the knowledge of the structure of $\mathcal{C}\ell(\mathcal{O})$ to be secure. The same holds if the original Couvegnies-Stolbunov-Rostovtsev's scheme is substituted with the CSIDH protocol.

Unfortunately, for primes p of cryptographic size, computing the structure of the ideal class group $\mathcal{C}\ell(\mathcal{O})$ is prohibitive. Therefore, alternative solutions to make the signature scheme safe have been proposed (see [15]), but none of them was satisfactory in terms of efficiency. Finally, in 2019 Beullens *et al.* [6] made a record class group computation which allowed to determine the structure of $\mathcal{C}\ell(\mathcal{O})$ for one set of CSIDH parameters, named CSIDH-512. As a result, they were able to design the first practical isogeny-based digital signature, named CSI-FiSh.

For CSIDH-512, $\mathcal{C}\ell(\mathcal{O})$ is a cyclic group, for which a generator $\mathfrak{g}$ is known. Therefore, it is isomorphic to $\mathbb{Z}_N$, where N is its order. This allows to uniquely represent each vector $(e_1, \ldots, e_n)$ with an integer in $\{0, \ldots, N\}$. Such unique representation guarantees that no information is leaked in the signature. However, the verification algorithm requires to compute the action of an element $[\mathfrak{g}]^a$, with $a \in \mathbb{Z}_N$. In order to make this computation feasible, it is necessary to find a vector $(f_1, \ldots, f_n)$ with *small* integral coordinates such that $\mathfrak{g}^a = \prod_{i=1}^{n} [\mathfrak{J}_i]^{f_i}$. In [6], an efficient lattice-based strategy to find a vector of this kind was proposed. We will review this strategy while also providing some heuristic results.

1.2. Roadmap

The rest of the paper is organized as follows. In Section 2 some theoretical results on elliptic curves and isogenies are reviewed. Section 3 describes the CSIDH key-exchange protocol and details its original implementation. In Section 4 we discuss the techniques that have

been proposed so far to speed up the execution of CSIDH. In Section 5 some of the methodologies for a constant-time implementation of CSIDH are described. Section 6 focuses on the CSIDH-512 set of parameters and its use for isogeny-based digital signature schemes. Finally, in Section 7 we draw some conclusions.

2. Preliminaries

In this section we recall some basic notions and results about elliptic curves over finite fields and isogenies. For more details, we refer the interested reader to [14, 19, 29, 38].

Let K be a field, that, throughout this work, we will always have $\mathrm{char}(K) > 3$. An affine Weierstrass equation over K is an equation of the form:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

with $a_1, a_3, a_2, a_4, a_6 \in K$. If the equation is non singular (i.e. $\frac{\partial E}{\partial X}(P) \cdot \frac{\partial E}{\partial Y}(P) \neq 0$ for all pairs $P \in \overline{K} \times \overline{K}$[1] satisfying equation (1)), it defines an elliptic curve over K. Given a second elliptic curve E' over K

$$E' : y^2 + a'_1xy + a'_3y = x^3 + a'_2x^2 + a'_4x + a'_6,$$

we say that E and E' are isomorphic over K (or $K-$isomorphic) if E' can be obtained from E by a change of variables of the form

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} u^2 & 0 \\ u^2s & u^3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} r \\ t \end{pmatrix} \qquad \text{with } u \in K^*, r, s, t \in K$$

and dividing the resulting equation by u^6. Every elliptic curve E can be written in the normal form $E : y^2 = x^3 + ax + b$ up to isomorphisms over K.

A binary operation, for which the additive notation is usually used, can be defined over the set containing the points of an elliptic curve E and a formal point ∞, making it an abelian group. An isogeny $\varphi : E \to E'$ between two elliptic curves E, E' over K is a rational map which is furthermore a group homomorphism. We say that φ

1. $\overline{K}$ denotes the algebraic closure of the field K.

is separable if $\overline{K}(E)^2$ is a separable field extension of $\varphi^* \left(\overline{K}(E')\right)$.[3] Furthermore, the degree of φ is defined as the dimension of $\overline{K}(E)$ over $\varphi^* \left(\overline{K}(E')\right)$, and we denote it by $\deg(\varphi)$. For a separable isogeny φ, the relation $\deg(\varphi) = |\mathrm{Ker}(\varphi)|$ holds [41, Thm. 12.8].

An endomorphism of an elliptic curve E is an isogeny from E to E. We denote by $\mathsf{End}(E)$ the set containing all endomorphisms of E and the zero map. This set is a ring with respect to pointwise addition and composition. An elliptic curve E is called supersingular if $\mathsf{End}(E)$ is non commutative, otherwise it is called ordinary.

Despite some of the results discussed below hold for general fields, for the sake of simplicity we will restrict the treatment to the case $K = \mathbb{F}_p$, where p is a prime, until the end of the section.

An elliptic curve $E : y^2 = x^3 + ax + b$ over $\mathbb{F}_p$ is supersingular if and only if the subgroup of its rational points, i.e.

$$E(\mathbb{F}_p) := \left\{ (x_0, y_0) \in \mathbb{F}_p \times \mathbb{F}_p \middle|\ y_0^2 = x_0^3 + ax_0 + b \right\} \cup \{\infty\}$$

has cardinality equal to $p + 1$ [42, Thm. 4.1]. Furthermore, given a supersingular elliptic curve E over $\mathbb{F}_p$, an elliptic curve E' over $\mathbb{F}_p$ is isogenous to E if and only if E' is supersingular [40, §3].

By definition, each of the coordinates of an isogeny $\varphi : E \to E'$ is the fraction of two polynomials in $\overline{\mathbb{F}}_p[x, y]$; if their coefficients lie in $\mathbb{F}_p$, then φ is said to be defined over $\mathbb{F}_p$. We denote by $\mathsf{End}_{\mathbb{F}_p}(E)$ the subring of $\mathsf{End}(E)$ containing all endomorphisms of E defined over $\mathbb{F}_p$. The ring $\mathsf{End}_{\mathbb{F}_p}(E)$ is isomorphic to an order in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$ [16, Thm. 2.1]. In particular, we have either $\mathsf{End}_{\mathbb{F}_p}(E) \simeq \mathbb{Z}[\sqrt{-p}]$ or $\mathsf{End}_{\mathbb{F}_p}(E) \simeq \mathcal{O}_{\mathbb{Q}(\sqrt{-p})}$. In the former case we say that E lies on the floor, otherwise we say that E lies on the surface. CSIDH considers only supersingular elliptic curves on the floor, and works with primes p of a specific form. More precisely, the primes p used by CSIDH are of the form $p = 4\ell_1 \cdot \ldots \cdot \ell_n - 1$, where $\ell_1, \ldots, \ell_n$ are distinct odd primes. In this case, the group $E(\mathbb{F}_p)$ of a supersingular elliptic curve E over $\mathbb{F}_p$ has two possible

2. $\overline{K}(E)$ denotes the fraction field of $\overline{K}[x, y]/(E)$, where E here detotes the bivariate polynomial $y^2 - x^3 - ax - b$.

3. φ^* is the map which sends $f \in \overline{K}(E')$ in $\varphi^*(f) = f \circ \varphi \in \overline{K}(E)$.

structures, namely:

$$\mathbb{Z}_4 \times \mathbb{Z}_{\ell_1} \times \ldots \times \mathbb{Z}_{\ell_n} \quad \text{or} \quad \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_{\ell_1} \times \ldots \times \mathbb{Z}_{\ell_n}.$$

Theorem 2.7 from [16] implies that E lies on the floor if the former relation holds. Therefore, the number of the 2-torsion points in $E(\mathbb{F}_p)$ locates the curve E.

The ideal class group $\mathcal{C}\ell(\mathcal{O})$ acts freely and transitively on $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, the set of all supersingular elliptic curves E, up to $\mathbb{F}_p-$isomorphisms, such that there exists an isomorphism between $\mathcal{O}$ and $\mathsf{End}_{\mathbb{F}_p}(E)$ mapping $\sqrt{-p}$ into the Frobenius endomorphism $\pi : (x, y) \mapsto (x^p, y^p)$ [36, Thm. 4.5]. The action of $\mathcal{C}\ell(\mathcal{O})$ is denoted by $\star$ and, given $[\mathfrak{a}] \in \mathcal{C}\ell(\mathcal{O})$ and $[E] \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, the element $[\mathfrak{a}] \star [E]$ will be also denoted by $E/\mathfrak{a}$ or $\mathfrak{a} \star E$ for simplicity. The CSIDH scheme exploits the action $\star$ of $\mathcal{C}\ell(\mathcal{O})$, with $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$, on $\mathcal{E}\ell\ell_p(\mathbb{Z}[\sqrt{-p}], \pi)$ to construct a key exchange.

In the CSIDH setting, to each small odd prime ℓ_i it corresponds an element $[\mathfrak{I}_i] \in \mathcal{C}\ell(\mathcal{O})$ whose action (and that of its inverse) can be efficiently computed. Indeed, $\mathfrak{I}_i$ is the fractional ideal $\mathfrak{I}_i = (\ell_i, \sqrt{-p} - 1)$, while $[\mathfrak{I}_i]^{-1}$ has $\bar{\mathfrak{I}}_i = (\ell_i, \sqrt{-p} + 1)$ as representative. For all $i = 1, \ldots, n$ and $[E] \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, the curves $E/\mathfrak{I}_i$ and $E/\bar{\mathfrak{I}}_i$ are the ranges of the separable isogenies originating from E and having kernel $E(\mathbb{F}_p) \cap E[\ell_i]$ and $\left\{P \in E(\mathbb{F}_{p^2}) \,\middle|\, \pi(P) = -P\right\} \cap E[\ell_i]$, respectively. The images $E/\mathfrak{I}_i$ and $E/\bar{\mathfrak{I}}_i$ are uniquely defined, modulo isomorphisms, by their kernel [38, § III, Thm.4.1]. Furthermore, they can be explicitly computed from a point P, lying on $E(\mathbb{F}_p)$ or $E(\mathbb{F}_{p^2})$ and having order divisible by ℓ_i, by means of Velu's formulas [30].

2.1. Montgomery Curves

Let $A \neq \pm 2$ be an element of the field $\mathbb{F}_p$. A Montgomery curve $E_{A,1}$ over $\mathbb{F}_p$ is an elliptic curve with equation[4] $E_{A,1} : y^2 = x^3 + Ax^2 + x$. The following theorem shows how, in the CSIDH setting, Montgomery curves allow to represent each element of $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ with a single element of the base field $\mathbb{F}_p$ [8, Prop. 8].

4. The notation $E_{A,1}$ is due to a simplification we are making in the definition. Indeed, to be accurate, a Montgomery curve is an affine plane curve defined by an equation of the form $E_{A,B} : By^2 = x^3 + Ax^2 + x$, where $B \neq 0$. However, since throughout the paper we will deal only with Montgomery curves having $B = 1$, for the sake of simplicity we decided to slightly deviate from the standard terminology.

Theorem 1. Let $p > 3$ be a prime such that $p \equiv 3 \pmod 8$, and E a supersingular elliptic curve over $\mathbb{F}_p$. Then, E lies on the floor if and only if there exists $A \in \mathbb{F}_p$ such that E is $\mathbb{F}_p$-isomorphic to the Montgomery curve $E_{A,1} : y^2 = x^3 + Ax^2 + x$. Moreover, if such A exists, it is unique.

Since $[E] \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ is the class of supersingular elliptic curves that are $\mathbb{F}_p$-isomorphic to E, all the curves therein are $\mathbb{F}_p$-isomorphic to the same Montgomery curve $E_{A,1}$. This makes it possible to identify $[E]$ with A.

Montgomery curves also offer explicit formulas for the computation of the action of the special elements $[\mathfrak{I}_1], \ldots, [\mathfrak{I}_n]$ (see [12, Thm. 1] and [26, Thm.1]).

Theorem 2. Consider a point $P \in E(\overline{\mathbb{F}_p}) \setminus \{\infty\}$ of order $\ell = 2d + 1$ on the Montgomery curve $E_{A,1} : y^2 = x^3 + Ax^2 + x$ defined over $\mathbb{F}_p$. Let $\sigma = \sum_{i=1}^{d} x_{[i]P}, \tilde{\sigma} = \sum_{i=1}^{d} (1/x_{[i]P})$ and $\omega = \prod_{i=1}^{d} x_{[i]P}$, where $x_{[i]P}$ is the $x-$coordinate of the point $[i]P$. Then, the Montgomery curve

$$E_{A',1} : y^2 = x^3 + A'x^2 + x$$

with

$$A' = (6\tilde{\sigma} - 6\sigma + a) \cdot \omega^2,$$

is the codomain of the separable isogeny $\phi : E_{A,1} \to E_{A',1}$ of degree ℓ and kernel $\langle P \rangle$, whose coordinates are given by the map

$$\phi : (x, y) \mapsto (f(x), \omega y f'(x))$$

where

$$f(x) = x \cdot \prod_{i=1}^{d} \left(\frac{x \cdot x_{[i]P} - 1}{x - x_{[i]P}} \right)^2$$

and $f'(x)$ is its derivative.

3. CSIDH

3.1. The key-exchange Protocol

As anticipated, the isogeny-based key exchange CSIDH [8] is obtained from the action $\star$ of $\mathcal{C}\ell(\mathcal{O})$ on $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, with $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$ for

a fixed CSIDH prime $p = 4\ell_1 \cdot \ldots \cdot \ell_n - 1$. Since computing the action of a random element $[\mathfrak{a}] \in \mathcal{C}\ell(\mathcal{O})$ has exponential complexity, the scheme restricts itself to actions of elements of the form $\prod_{i=1}^{k} [\mathfrak{I}_i]^{e_i}$, where the integral exponents e_i are chosen from some small interval $[-B, B]$. Indeed, computing these actions corresponds to calculating sequences of efficient actions and, in particular, of isogenies, as we saw in the previous section.

In addition to the prime p (which, in turn, determines n), the public parameters of the CSIDH protocol specify the value of the small positive integer B and a starting supersingular elliptic curve over $\mathbb{F}_p$ lying on the floor. It is standard to set this curve as $E_0 : y^2 = x^3 + x$ (defined over $\mathbb{F}_p$), which is supersingular and lies on the floor [41, Thm. 4.37].

The set of private keys is $[-B, B]^n$: a vector $(e_1, \ldots, e_n)$ represents the element $[\mathfrak{a}] = \prod_{i=1}^{n} [\mathfrak{I}_i]^{e_i}$ of $\mathcal{C}\ell(\mathcal{O})$. $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ is the set of public keys, and the public key corresponding to the private key $(e_1, \ldots, e_n)$ is defined as $E_0/\mathfrak{a}$.

Below we recall how the key exchange between two users, Alice and Bob, works. At the beginning, Alice and Bob have the key pairs $\left((e_1, \ldots, e_n), {}^{E_0}\!/_{\mathfrak{a}}\right)$ and $\left((f_1, \ldots, f_n), {}^{E_0}\!/_{\mathfrak{b}}\right)$, respectively. Here $\mathfrak{a} = \prod_{i=1}^{n} [\mathfrak{I}_i]^{e_i}$ and $\mathfrak{b} = \prod_{i=1}^{n} [\mathfrak{I}_i]^{f_i}$. Then the interaction proceeds as follows:

— Alice and Bob exchange their public keys;
— Alice computes $[\mathfrak{a}] \star {}^{E_0}\!/_{\mathfrak{b}} = [\mathfrak{a}][\mathfrak{b}] \star [E_0]$;
— Bob computes $[\mathfrak{b}] \star {}^{E_0}\!/_{\mathfrak{a}} = [\mathfrak{b}][\mathfrak{a}] \star [E_0]$.

The commutativity of $\mathcal{C}\ell(\mathcal{O})$ guarantees that both users obtain the same element of $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, which, thanks to Theorem 1, can be represented by a single element in $\mathbb{F}_p$. Furthermore, Theorem 2 instructs how to compute the action of $[\mathfrak{I}_i]$ on an element $[E_{A,1}] \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$. In particular, a point $P \in E_{A,1}(\mathbb{F}_p)$ having order ℓ_i is used to compute the isogeny φ which emanates from $E_{A,1}$, has $< P >$ as kernel, and whose range is exactly ${}^{E_{A,1}}\!/_{\mathfrak{I}_i}$. We note that the point $\varphi(P)$ has order $\mathsf{ord}(P)/\ell_i$ and therefore can potentially be used to compute the action of another element $[\mathfrak{I}_j]$, with $j \neq i$. Both ${}^{E_{A,1}}\!/_{\mathfrak{I}_i}$ and $\varphi(P)$ can be calculated using the formulas in Theorem 2.

Also the action of an element $[\mathfrak{I}_i]^{-1}$ can be computed working on the base field $\mathbb{F}_p$. Let us consider a Montgomery curve $E_{A,1}$ representing an element in $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$. Since the action $\star$ is free and transitive, there exists a unique $[\mathfrak{a}]$ in $\mathcal{C}\ell(\mathcal{O})$ such that $[E_{A,1}] = [\mathfrak{a}] \star [E_0]$. It is possible to show that $E_{-A,1}$ is the Montgomery curve representing $[\mathfrak{a}]^{-1} \star [E_0]$ (see [8, Remark 5]) . As a consequence, the result of the action of $[\mathfrak{I}_i]^{-1}$ on $[E_{A,1}]$ is $[E_{-A',1}]$, where $[E_{A',1}] = [\mathfrak{I}_i] \star [E_{-A,1}] = ([\mathfrak{I}_i][\mathfrak{a}]^{-1}) \star [E_0]$. Indeed we have:

$$[\mathfrak{I}_i]^{-1} \star [E_{A,1}] = ([\mathfrak{I}_i]^{-1}[\mathfrak{a}]) \star [E_0] = ([\mathfrak{I}_i][\mathfrak{a}]^{-1})^{-1} \star [E_0].$$

A random $x \in \mathbb{F}_p$ is the x-coordinate of a point P in $E_{A,1}(\mathbb{F}_p)$ or its opposite $-x$ is the x-coordinate of $P \in E_{-A,1}(\mathbb{F}_p)$, depending if $x^3 + Ax^2 + x$ is a quadratic residue modulo p or not[5]. In the former case, P can be used to produce a point in $E_{A,1}(\mathbb{F}_p)$ having order ℓ_i, for some $i \in \{1, \ldots, n\}$, and hence compute the action of $[\mathfrak{I}_i]$. In the latter case, P can be used to produce a point in $E_{-A,1}(\mathbb{F}_p)$ having order ℓ_i, for some $i \in \{1, \ldots, n\}$, and compute the action of $[\mathfrak{I}_i]$ on $[E_{-A,1}]$, from which the action of $[\mathfrak{I}_i]^{-1}$ on $[E_{A,1}]$ is then easily deduced.

The complete procedure to compute the action of a vector $(e_1, \ldots, e_n)$ on $[E_0]$ or a public key $[E_{A,1}]$ is detailed in Algorithm 1, which was firstly depicted in the original CSIDH paper. In the following pages, we will refer multiple times to this algorithm, since nearly all CSIDH optimisations build on it.

4. Speed-up of the class group computation

Despite providing a practical methodology to calculate the action of a special element $[\mathfrak{I}_i] \in \mathcal{C}\ell(\mathcal{O})$, the computation of the formulas from Theorem 2 is slowed down by the several inversions in $\mathbb{F}_p$ they require. A natural way to avoid inversions is using the projective arithmetic.

5. Note that every CSIDH prime p is congruent to 3 modulo 4, an so -1 is not a quadratic residue modulo p.

Algorithm 1 Evaluating the class group action

Input $A \in \mathbb{F}_p$ s.t. $[E_{A,1} : y^2 = x^3 + Ax^2 + x] \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, $(e_1, \ldots, e_n) \in \mathbb{Z}^n$.
Output $A' \in \mathbb{F}_p$ such that $[E_{A',1}] = [\mathfrak{I}_1]^{e_1} \cdot \ldots \cdot [\mathfrak{I}_n]^{e_n} \star [E_{A,1}]$.

```
while some e_i ≠ 0 do
    Sample a random x ∈ F_p.
    iso ← 0
    Set s ← 1 if x^3 + Ax^2 + x is a square in F_p, else s ← −1.
    S ← {i | sign(e_i) = s}
    if S ≠ ∅ then
        Set P as a point whose abscissa is s · x.
        k ← ∏_{i∈S} ℓ_i
        P ← [(p+1)/k] P
        for i ∈ S do
            K ← [k/ℓ_i] P
            if K ≠ ∞ then
                Compute φ : E_{s·A,1} → E_{A',1}, Ker(φ) =< K >.
                A ← A', P ← φ(P), iso ← 1
                k ← k/ℓ_i, e_i ← e_i − s
            end if
        end for
        if iso == 1 then
            A ← s · A', iso ← 0
        end if
    end if
end while
return A
```

4.1. Projective arithmetic

Given a Montgomery curve $E_{A,1} : y^2 = x^3 + Ax^2 + x$ over a field K, its projective form is given by the equation:

$$E^*_{A,1} : Y^2Z = X^3 + AX^2Z + XZ^2.$$

We recall that to every point $(x, y) \in E_{A,1}$, it corresponds a point in $E^*_{A,1}$, i.e. $[x : y : 1]$. Conversely, to a point $[u : v : z] \in E^*_{A,1}$ with $z \neq 0$, it corresponds $(u/z, v/z) \in E_{A,1}$. The unique point $[u : v : z]$ in $E^*_{A,1}$ having $z = 0$ is equal to $[0 : 1 : 0]$, and it corresponds to the point at infinity ∞ in $E_{A,1}$. This bijection allows to extend the

group law of $E_{A,1}$ to $E^*_{A,1}$.

It is possible to projectivise Theorem 2 as follows. Consider a Montgomery curve $E_{A,1} : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$, and a point $K \in E_{A,1}(\mathbb{F}_p)$ of order $\ell = 2d + 1$. We denote by φ the isogeny originating from $E_{A,1}$ and having $< K >$ as kernel. Let $[x_i : y_i : z_i]$, $[x : y : z]$ and $[x' : y' : z']$ be the coordinates of the projective points corresponding to $[i]K$, P and $\varphi(P)$, respectively. Then we have [12]:

$$x' = x \left(\prod_{i=1}^{d} (x \cdot x_i - z \cdot z_i) \right)^2 , \quad z' = z \left(\prod_{i=1}^{d} (x \cdot z_i - x_i \cdot z) \right)^2 \quad (2)$$

The above formulas were already exploited in [8], where, after the computation of the values of x_i and z_i for $i = 1, ..., d$, $x \cdot x_i - z \cdot z_i$ and $x \cdot z_i - x_i \cdot z$ are evaluated at the cost of $4d$ field multiplications and $2d$ field additions.

Meyer and Reith [28] highlighted the possibility of trading $2d$ of the field multiplications for $2d + 2$ extra field additions, obtaining an efficiency improvement. This is obtained by replacing the formulas in Equation 2 with

$$x' = x \left(\prod_{i=1}^{d} [(x - z)(x_i + z_i) + (x + z)(x_i - z_i)] \right)^2 \quad (3)$$

$$z' = z \left(\prod_{i=1}^{d} [(x - z)(x_i + z_i) - (x + z)(x_i - z_i)] \right)^2 \quad (4)$$

In [28], a speed up for the computation of the image curve in Theorem 2 was also proposed. It is derived from the correspondence between Montgomery curves and a family of special affine plane curves, named twisted Edwards curves. At a high level, the idea is to determine the twisted Edwards curve corresponding to $E_{A,1}$, then compute the *isogeny* between twisted Edwards curves corresponding to the action of $[\mathfrak{J}_i]$, and finally revert its range to the corresponding Montgomery curve, which is exactly $E_{A',1}$. This procedure saves some field multiplications, and is particular effective for the largest values of i.

A recent work of Bernestein *et al.* [3] focused on both problems considered in this section, i.e. computing the image curve of an

isogeny φ and evaluate φ on a point, introducing a new efficient baby-step giant-step evaluation technique for rational functions over $\mathbb{F}_p$.

4.2. Collisions in the secret-key space

In CSIDH, the space of private keys is the set $[-B, B]^n$, where B is a small positive integer for which the inequality $(2B+1)^n \geq |\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])|$ holds. In [8], some heuristic arguments are presented to corroborate the assumption that this choice suffices to cover the entire ideal class group $\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])$. However, a priori, collisions (i.e. two distinct vectors $(e_1, \ldots, e_n)$, $(e'_1, \ldots, e'_n) \in [-B, B]^n$ such that $[\mathfrak{I}_1^{e_1} \cdot \ldots \cdot \mathfrak{I}_n^{e_n}] = [\mathfrak{I}_1^{e'_1} \cdot \ldots \cdot \mathfrak{I}_n^{e'_n}]$) may happen.

Onuki *et al.* [33] provided a class of them, showing that collisions actually exist. Indeed, the element $[\mathfrak{I}_1 \cdot \ldots \cdot \mathfrak{I}_n]$ has order 3 in $\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])$ [33, Thm. 3]. Therefore, the vectors $(e_1, \ldots, e_n)$, $(e_1 + 3, \ldots, e_n + 3)$ and $(e_1 - 3, \ldots, e_n - 3)$ are representative of the same ideal class. In the light of this, the space of private keys $[-B, B]^n$ has cardinality less than $(2B+1)^n$ if $B \geq 3$. This issue can be avoided using $[-B_1, B_1] \times \ldots \times [-B_n, B_n]$ as space of private keys, where at least one of the B_i's is less than 3.

The element $[\mathfrak{I}_1 \cdot \ldots \cdot \mathfrak{I}_n]$ not only has order 3, but the computation of its action and that of its inverse $[\overline{\mathfrak{I}_1} \cdot \ldots \cdot \overline{\mathfrak{I}_n}]$ is especially efficient [33, Thm. 7] and they can be exploited while calculating the action of elements $(e_1, \ldots, e_n)$ having Hamming weight equal to n.

Theorem 3. Let $A \in \mathbb{F}_p$ such that $[E_{A,1}] \in \mathcal{E}\ell\ell_p(\mathbb{Z}[\sqrt{-p}], \pi)$. Then

$$[\mathfrak{I}_1 \cdot \ldots \cdot \mathfrak{I}_n] \star [E_{A,1}] = [E_{A',1}], \quad [\overline{\mathfrak{I}_1} \cdot \ldots \cdot \overline{\mathfrak{I}_n}] \star [E_{A,1}] = [E_{A'',1}]$$

where

$$A' = -2\frac{A+6}{A-2}, \quad A'' = 2\frac{A-6}{A+2}.$$

As we already observed, a subset $\mathcal{A}$ of $\mathbb{Z}^n$ covering the entire ideal class group $\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])$ should not necessarily be of the form $[-B, B]^n$ for some positive integer B. In [31], Nakagawa *et al.* considered the problem of determining an optimal subset $\mathcal{A}_{opt} \subset \mathbb{Z}^n$ which minimises $\mathbb{E}_{\mathrm{e} \in \mathcal{A}}[T(\mathrm{e})]$ under the condition $|\mathcal{A}| \geq |\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])|$. Here, $T(\mathrm{e})$ denotes the number of field multiplications required to

compute the action related to $e \in \mathbb{Z}^n$, while $\mathbb{E}_{e \in \mathcal{A}}[T(e)]$ denotes the expected value of $T(e)$ taken over a uniform choice of $e \in \mathcal{A}$.

4.3. Optimal key space

In [31], a procedure to tackle the above mentioned optimisation problem is presented. In particular, under the assumption that $\mathcal{A}_{opt}$ can be written as $\{e \in \mathbb{Z}^n | T(e) \leq r\}$ for some $r \in \mathbb{N}$, the procedure computes an approximate lower bound for all r such that $\{e \in \mathbb{Z}^n | T(e) \leq r\}$ has cardinality greater than or equal to $|\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])|$. Nakagawa *et al.* provided a concrete value for the lower bound of r only for one specific set of CSIDH parameters, that we are going to encounter several times in the following pages. This set of parameters is called CSIDH-512 and it specifies a 512-bit prime p of the form $p = 4 \cdot \ell_1 \cdot \ldots \cdot \ell_{74} - 1$, where ℓ_1 through ℓ_{73} are the first 73 odd primes and $\ell_{74} = 587$. In this case the value for the lower bound r_{opt} of r is 336428, while the average computational cost $T(e)$ in $\mathcal{A}_{opt} = \{e \in \mathbb{Z}^n | T(e) \leq r_{opt}\}$ is $T_{opt} \sim 332000$.

However, uniformly sampling from the proposed optimal key space $\mathcal{A}_{opt}$ is very hard. Therefore, in [31] a "transposition" of this set is provided. To be more precise, it is proved that the average value of $T(e)$ for e varying in the L_1-ball with radius 152 is 355804, which is only 1.08 times bigger than T_{opt}. Hence, it is possible to replace $\mathcal{A}_{opt}$ with this L_1-ball, in order to be able to easily sample from the secret-key space while retaining (almost) the same efficiency of the optimal secret-key space.

4.4. CSIDH on the Surface

The CSIDH protocol and its subsequent optimisations fix a prime p of a suitable form and work with the supersingular elliptic curves over $\mathbb{F}_p$ that lie on the floor. This choice is due to the possibility of representing each element of $\mathcal{E}\ell\ell_p(\mathbb{Z}[\sqrt{-p}], \pi)$ with a unique element of the base field $\mathbb{F}_p$ (see Theorem 1).

In [7], Castryck and Decru exhibited a new elliptic curve model that allows to transpose the above mentioned property, and consequently the CSIDH scheme, also to supersingular elliptic curves on the surface. Moreover, they proposed a new 512-bit prime and a suitable secret-key space which produced a speed up of about 5.68% with re-

spect to CSIDH instantiated with the CSIDH-512 set of parameters[6]. The proposed prime is $p = 2^3 \cdot 3 \cdot \ell_1 \cdot \ldots \cdot \ell_{74} - 1$, where the ℓ_i's are the 74 consecutive primes from 3 to 389 skipping 347 and 357, while the secret keys are sampled from $[-137, 137] \times [-4, 4]^3 \times [-5, 5]^{46} \times [-4, 4]^{25}$. The result on how to associate a unique element of $\mathbb{F}_p$ to each class in $\mathcal{E}\ell\ell_p\left(\mathcal{O}_{\mathbb{Q}(\sqrt{-p})}, \pi\right)$ [7, Prop. 4] is stated in the following theorem, which concludes the section.

Theorem 4. Let p be a prime such that $p \equiv 7 \pmod 8$, and E a supersingular elliptic curve over $\mathbb{F}_p$. Then, E lies on the surface if and only if there exists $A \in \mathbb{F}_p$ such that E is $\mathbb{F}_p$-isomorphic to the elliptic curve $y^2 = x^3 + Ax^2 - x$. Moreover, if such A exists, it is unique.

5. Constant-time class group computation

In analogy with the case of the classical Diffie-Hellman key-exchange protocol, it is standard to assume that the only way to break the CSIDH scheme is to recover one of the private keys. A passive attacker is only given the corresponding public keys, and therefore they have to solve an instance of the Group Action Inverse Problem (GAIP, in short), which is defined as follows.

Definition 1 (Group Action Inverse Problem (GAIP)). Let $[E_0]$ be an element in $\mathcal{E}\ell\ell_p([\mathbb{Z}(\sqrt{-p}], \pi)$, where $p \geq 5$ is a CSIDH prime. Given $[E]$ sampled uniformly at random from $\mathcal{E}\ell\ell_p(\mathbb{Z}[\sqrt{-p}], \pi)$, the GAIP_p problem consists in finding the unique element $[\mathfrak{a}] \in \mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])$ such that $[\mathfrak{a}] \star [E_0] = [E]$.

The best known classical algorithm to solve the GAIP_p problem has time complexity $O(\sqrt{N})$, where $N = |\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])| \sim \sqrt{p}$, while the best known quantum algorithm is Kuperberg's algorithm for the hidden shift problem [23, 24]. The latter has a subexponential complexity, but its concrete estimate is still an active area of research [34].

The original implementation of CSIDH (and those of the subsequent improvements) allows an active attacker to obtain some information

6. Note that the comparison was made with the original CSIDH implementation, that does not use the improvements we saw in Section 4.1.

about the private keys analyzing the running time of Algorithm 1, since different private keys require a different number of isogeny computations. Two possible strategies to avoid this leakage have been proposed so far. In the first one (see [5]), the computational cost for the action of any element $(e_1, \ldots, e_n)$ in $[-B, B]^n$ is always the same, independently of $(e_1, \ldots, e_n)$ and the randomness used; this strategy has a negligible failure probability. In the second strategy (see [27]), the computational cost for the action of a vector $(e_1, \ldots, e_n)$ is made independent of $(e_1, \ldots, e_n)$ itself, but might vary due to the used randomness. In the following, we will label as constant time the resulting implementations of the two strategies, even though they aim at substantially different goals.

In the second strategy, for each prime ℓ_i in the definition of p, the number of ℓ_i-isogenies to be computed for each vector $(e_1, \ldots, e_n)$ in $[-B, B]^n$ is fixed to the maximum value, i.e. B. In other words, if $|e_i| < B$, then $B - |e_i|$ extra artificial isogenies of degree ℓ_i are computed. In particular, for each of these artificial isogenies, after obtaining a point P of order ℓ_i, the scalar multiplication $[\ell_i]P$ is executed with the aim of mimic the operations in Theorem 2, but the curve parameters are left unchanged. This means that an artificial isogeny is not a real isogeny, but just a set of operations having the same cost of those necessary to compute a proper ℓ_i-isogeny. Since these extra isogenies are useless for the computation of the action of $(e_1, \ldots, e_n)$, they are called *dummy isogenies* [28].

However, the computational costs for the actions of two vectors $(e_1, \ldots, e_n)$, $(e'_1, \ldots, e'_n)$ might differ even when $|e_i| = |e'_i|$ for every $i \in \{1, \ldots, n\}$, and therefore even when computing dummy isogenies. For example, Meyer *et al.* noted in [28, §6] that the running time for computing the action of $(5, 5, 5, \ldots, 5)$[7] when working with the CSIDH-512 set of parameters set is higher than the running time for computing the action of $(5, -5, 5, -5, \ldots, 5, -5)$. The reason for this discrepancy is that, once a random x-coordinate is sampled (line 2 of Algorithm 1), it can be used only to compute isogenies for those indices belonging to the set S (line 5 of Algorithm 1). For the first vector, when $s = -1$, the set S is always empty - and so the computed x cannot be used - while when $s = 1$ the set S can contain up to n elements, making the scalar multiplications in lines

7. They did not use the improvement described in Section 4.2, otherwise, using Theorem 3, the computation would have been trivial.

8 and 10 of Algorithm 1 unbalanced and therefore more expensive.

In the light of the above observation, in order to make the dummy-isogenies strategy work, Meyer *et al.* proposed in [27] to replace the symmetric set of private keys $[-B, B]^n$ with the asymmetric set $[0, 2B]^n$, since this choice does not affect the hardness assumption on the GAIP problem.

We notice that, after the computation of one dummy isogeny, the curve coefficient is not changed, contrarily to what happens with a real isogeny (line 14 of Algorithm 1). Hence, an active attacker having cache access during the execution of class group actions could determine the number of dummy computations. This issue is avoided by multiplying the (projective) coefficient $[A : C]$ of the curve by a random $\alpha \in \mathbb{F}_p^*$ after each dummy isogeny [27, §5.1].

After this general overview on the different strategies for achieving a constant-time implementation of the CSIDH scheme, we provide more technical details about them in the following sections.

5.1. Elligator

The constant-time implementation of CSIDH presented in [5] uses an efficient method to generate random rational points on a Montgomery curve $E_{A,1} : y^2 = x^3 + Ax^2 + x$, defined over $\mathbb{F}_p$ and such that $[E_{A,1}] \in \mathcal{E}\ell\ell_p(\mathbb{Z}[\sqrt{-p}], \pi)$, or on its counterpart $E_{-A,1}$. In particular, the method exploits the fact that, given $x \in \mathbb{F}_p$ and $y = (x^3 + Ax^2 + x)^{\frac{p+1}{4}}$,

$$y^4 = (x^3 + Ax^2 + x)^{p+1} = (x^3 + Ax^2 + x)^2.$$

Then $y^2 = \pm(x^3 + Ax^2 + x)$ and, consequently, (x, y) is a point of $E_{A,1}$ when $y^2 = x^3 + Ax^2 + x$, otherwise $(-x, y)$ is a point of the curve $E_{-A,1}$ used to compute the action of elements $[\mathfrak{J}_i]^{-1}$ (see Section 3.1).

The method to sample rational points in $E_{A,1}$ or in $E_{-A,1}$ which results from the above observation is depicted in the following Algorithm 2. It is called *Elligator* 2 *map*, and it was designed by Bernstein *et al.* in [4].

Algorithm 2 Elligator 2 map

Input: The coefficient $A \in \mathbb{F}_p^*$ of a Montgomery curve $E_{A,1}$ such that $[E_{A,1}] \in \mathcal{E}\ell\ell_p(\mathbb{Z}[\sqrt{-p}], \pi)$, $s \in \{1, -1\}$.
Output: x-coordinate of a rational point in $E_{A,1}$ if $s = 1$, of a rational point in $E_{-A,1}$ if $s = -1$.

1: Uniformly sample an element u from $\{2, \ldots, \frac{p-1}{2}\}$.
2: $v \leftarrow \frac{A}{u^2-1}$
3: Set e as the Legendre Symbol of $v^3 + Av^2 + v$ over p.
4: if $e == s$ then
5: return v
6: else
7: return $v + A$
8: end if

The Elligator 2 map can be extended to the case $A = 0$, i.e. the case where $E_{A,1}$ is the curve E_0, setting $v = u$ (instead of $v = \frac{A}{u^2-1}$). However, the use of the Elligator 2 map at the beginning of Algorithm 1 does not seem to be convenient, since some full order points of E_0 can be pre-computed offline.

We observe that, given a uniformly random rational point $P \in E_{A,1}(\mathbb{F}_p)$, the probability that its order is not divisible by ℓ_i is $1/\ell_i$. Moreover, among the $(p-3)/2$ points that could be sampled by the Elligator 2 map, at most $(p+1)/\ell_i$ have order not divisible by ℓ_i. Then, the probability of sampling a point with order not divisible by ℓ_i using the Elligator 2 map is upper bounded by

$$\left(\frac{2}{\ell_i}\right)\frac{p+1}{p-3} \sim \frac{2}{\ell_i}.$$

In [5], it is claimed that the heuristic probability is almost exactly $1/\ell_i$, so that the distribution of points output by the Elligator 2 map is heuristically close to the uniform one.

The constant-time implementation of CSIDH which relies on the Elligator 2 map is detailed in the following Algorithm 3, which is a simplified version of Algorithm 6.1 from [5]. For every $i \in \{1, \ldots, n\}$, r steps (for some positive integer r) are executed, each of them having the same computational cost. If, after the r steps, the computation of the action of $[\mathfrak{I}_i]^{e_i}$ is incomplete, the algorithm fails. The parameter r is tuned to increase/decrease the success probability of the algorithm.

Algorithm 3 Constant-time class group action evaluation

Input: The coefficient $A \in \mathbb{F}_p^*$ of a Montgomery curve $E_{A,1}$ s.t. $[E_{A,1}] \in \mathcal{E}\ell\ell_p(\mathbb{Z}[\sqrt{-p}], \pi)$, $(e_1, \ldots, e_n) \in [-B, B]^n$.
Output: $A' \in \mathbb{F}_p$ s.t. $[E_{A',1}] = [\prod_{i=1}^{n} \mathfrak{I}_i^{e_i}] \star [E_{A,1}]$.

1: for $i \in \{1, \ldots, n\}$ do
2: for $j \in \{1, \ldots, r\}$ do
3: $s \leftarrow \mathsf{sign}(e_i)$
4: Determine a point $P \in E_{s \cdot A,1}$ using the Elligator 2 map
5: $Q \leftarrow [\frac{p+1}{\ell_i}]P$
6: if $Q \neq \infty$ then
7: Compute $\varphi : E_{s \cdot A,1} \rightarrow E_{A',1}$, $\mathrm{Ker}(\varphi) =< Q >$
8: if $s \neq 0$ then
9: $A \leftarrow s \cdot A'$
10: end if
11: $e_i \leftarrow e_i - s$
12: end if
13: end for
14: end for
15: if $(e_1, \ldots, e_n) == (0, \ldots, 0)$ then
16: return A
17: else
18: return failure
19: end if

In terms of efficiency, the most expensive step in Algorithm 2 is the computation of $A/(u^2 - 1)$ (line 2), since it requires a field inversion. In [27, §5.3] Meyer *et al.* suggested the pre-computation of $1/(u^2 - 1)$ for 10 values of $u \in \{2, \ldots, \frac{p-1}{2}\}$. They used this variant of the Elligator 2 map together with dummy isogenies and the asymmetric key-space $[0, 2B]^n$ to obtain a constant-time implementation of CSIDH whose running time is independent of the private key. Since their algorithm is no-failure, if the 10 precomputed values for the Elligator 2 map do not give rise to points that can be exploited for the isogenies it remains to compute, then their algorithm retreats to the original Elligator 2 map which samples random points. However, this step makes the computation for the action of a vector $(e_1, \ldots, e_n)$ not independent of the private key. Indeed, the time required by the Elligator 2 map (initially running the variant with pre-computations and then, if necessary, the original one) to produce suitable rational points on the current supersingular curve

$E_{A,1}$ depends on A, which itself depends on the secret key.

To fix this issue, Cervantes-Vázquez *et al.* [9, §3] proposed a projective Elligator 2 map which uses randomness while still avoiding inversions. We observe that this projective variant leaves unchanged the probability of finding a point with a fixed order.

5.2. Simba

In [27], Meyer *et al.* proposed a technique to speed up their constant time implementation. This technique consists in splitting the set $\{1, \ldots, n\}$ into subsets, and working separately within each of them. To explain the technique and its relevance, we take into consideration Algorithm 1. Despite being substantially different from the algorithm in [27] discussed in the previous section, the scalar multiplications the two algorithms perform are quite similar, and therefore Algorithm 1 is well suited for the exposition.

In line 5, the indices i for which it remains to compute some isogenies are gathered in a set S. The size of S affects the cost of the point multiplications $[(p+1)/k]P$, where $k = \prod_{i \in S} \ell_i$, and $[k/\ell_i]P$ (in lines 9 and 11), as the following examples show.

Suppose that after j iterations of the while loop, the input vector has been updated into a vector $(e_1, \ldots, e_n)$ having Hamming weight equal to n, i.e. none of its entries is zero. Therefore, $S = \{1, \ldots, n\}$, $k = (p+1)/4$, $(p+1)/k = 4$ and $k/\ell_i = (p+1)/4\ell_i$. This means that one point multiplication, i.e. $[(p+1)/k]P$, is really cheap, while the other one, i.e. $[k/\ell_i]P$, is extremely expensive, since the scalar factor has almost the same size of p. On the other hand, if $(e_1, \ldots, e_n)$ has Hamming weight equal to $\lfloor n/2 \rfloor$, the sizes of the two scalar factors could be more balanced. Suppose that S is equal to $\{2, 4, \ldots, h\}$, where $h = n$ if n is even, $h = n - 1$ otherwise. Then both scalar factors $[(p+1)/k]$, $[k/\ell_i]$ are approximately equal to $\sqrt{p}$. The computational cost for the two corresponding scalar multiplications would be, in this case, smaller than the previous example. Roughly, this is due to the fact that computing two *medium*-size scalar multiplications is more efficient than computing one *big*-size scalar multiplication.

In [27], Meyer *et al.* proposed to split the set $\{1, 2, \ldots, n\}$ into multiple batches $S_1, \ldots, S_m$, with $m < n$ being a positive integer,

in order to artificially recreate the conditions of the second example discussed above. This is obtained by introducing a for-loop (indexed by $j \in \{1, \ldots, m\}$) before the construction of S, and defining S as the set $S = \{i \in S_j | e_i > 0\}$ within each iteration of this loop.

The proposal by Meyer *et al.* gives rise to two natural questions: which is the optimal value for m? what is the optimal distribution of the indices in the subsets $S_1, \ldots, S_m$? For the CSIDH-512 set of parameters and setting $S_j = \{j, m + j, 2m + j, \ldots\}$, a heuristic analysis shows that the optimal choice for m is 5 (see [27, §5.3]).

We conclude this section noticing that, given an elliptic curve $E_{A,1}$, finding a rational point whose order is divisible by a small prime ℓ_i has low probability. Therefore, it is expected that, after a certain number of rounds, in every batch will remain a few indices i such that e_i is not zero[8]. The same argument that has led to the introduction of separate batches suggests that it is convenient to merge the batches at a certain point of the computation. The algorithm incorporating the subdivision in m batches and the merging of those batches after μ steps is named SIMBA$_{m,\mu}$.

The alternative private-key space in which e_i, for each $i \in \{1, \ldots, n\}$, belongs to a tailored interval $[0, B_i]$ instead of a fixed interval $[0, 2B]$ is also taken into consideration in [27].

5.3. CSIDH keeping two points

The use of an asymmetric key space was introduced to avoid the leakage of information due to the fact that a random x-coordinate determines the value of $s \in \{-1, 1\}$ and allows to compute only the isogenies corresponding to those elements of $(e_1, \ldots, e_n)$ whose sign coincides with s. An alternative remedy consists in exploiting the Elligator 2 map and the fact that it generates simultaneously points on both the elliptic curves $E_{A,1}$ and $E_{-A,1}$.

Building on this idea, Onuki *et al.* [32] introduced a new constant-time implementation of CSIDH (i.e. with a running time independent of the input vector $(e_1, \ldots, e_n)$), which works with a secret-key space whose vectors have both negative and positive entries. The implementation considered in [32] for benchmarking also uses dummy

8. Assuming that the small indices are distribuited across the batches

isogenies and SIMBA.

The implementation proposed by Onuki *et al.* is slightly faster, for the CSIDH-512 set of parameters, than the CSIDH implementation in [27]. In particular, the SIMBA parameters giving the most efficient implementation are $m = 3$ and $\mu = 8$. Furthermore, each secret exponent e_i is chosen from a bespoke interval $[-B_i, B_i]$, where the vector $(B_1, \ldots, B_n)$ is defined as follows:

$$[5,6,7,7,7,7,8,8,8,9,10,10,10,10,9,9,9,8,7,7,7,7,7,$$
$$7,7,7,7,7,7,7,6,6,6,6,6,5,5,5,5,5,5,5,4,4,4,4,4,4,$$
$$4,4,4,4,4,4,4,4,4,3,3,3,3,3,3,3,3,3,3,2,2,2,2,2,1].$$

5.4. Optimal Strategies

In [22, §4], De Feo *et al.* presented an *optimal strategy approach*[9] for computing isogenies within the isogeny-based SIDH scheme. Hutchison *et al.* [21] recently extended this approach to the constant-time implementations in [27] and [32] using linear-programming techniques. The result is an implementation for the CSIDH-512 set of parameters which is %5.06 faster than the 2-torsion points variant in [9] that adopts all the improvements enlisted so far (i.e., the point-evalutation speed up, the use of twisted Edwards curves, the projective Elligator map and the SIMBA algorithm).

Chi-Domínguez and Rodríguez-Henríquez independently extended the use of strategis to the CSIDH scheme [10]. They achieved a moderate speed-up with respect to the implementation proposed in [21], even without embracing the use of the SIMBA algorithm.

6. Computations in a class group with known structure

Three sets of CSIDH parameters have been proposed so far: CSIDH-512, CSIDH-1024 and CSIDH-1792 [8, 15]. Each set specifies the number n of primes $\ell_1, \ldots, \ell_n$ defining the prime p, and the prime p itself. The bit size of p is approximately 512 in CSIDH-512, 1024

9. The word "optimal" here refers to the number of field operations. as the goal of the approach is to minimise this number.

in CSIDH-1024 and 1792 in CSIDH-1792. To the prime p, it corresponds the order $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$ in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$. Then, the action of the ideal class group $\mathcal{C}\ell(\mathcal{O})$ on the supersingular elliptic curves over $\mathbb{F}_p$ lying on the floor is exploited to shape a key-exchange protocol, i.e. CSIDH itself.

The cardinality of $\mathcal{C}\ell(\mathcal{O})$, called class number of $\mathcal{O}$, is finite. The best known algorithms for the class number computation have sub-exponential complexity with respect to the discriminant of the quadratic field $\mathbb{Q}(\sqrt{-p})$. This makes the computation of the class numbers of the orders employed by CSIDH extremely heavy, if not infeasible.

The impracticality of computing the class number of a CSIDH order $\mathcal{O}$ has the consequence that the structure of the ideal class group $\mathcal{C}\ell(\mathcal{O})$ remains unknown. This scenario does not represent a substantial problem for CSIDH and its security[10], but it represents a major obstacle for the design of an efficient CSIDH-based digital signature scheme. As a result, a record class number computation for determining the structure of $\mathcal{O}$ for the CSIDH-512 set of parameters has been necessary to construct the first practical isogeny-based digital signature scheme, CSI-FiSh [6].

The class number computation executed by Beullens *et al.* in [6] determined that, for the prime p in the CSIDH-512 set of parameters, it holds:

$$\begin{aligned}\left|\mathcal{C}\ell\left(\mathbb{Z}\left[\sqrt{-p}\right]\right)\right| =& 3 \cdot 37 \cdot 1407181 \cdot 51593604295295867744293584889 \cdot \\ & 315994145046819958530082787455878322049 09 \\ & \sim 2^{257.136}.\end{aligned}$$

As a consequence, $\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])$ is a cyclic group[11], and it is showed that the element $[\mathfrak{I}_1] = [(3, \sqrt{-p} - 1)]$, which we denote by $\mathfrak{g}$, is a generator. Furthermore, the discrete logarithm of $[\mathfrak{I}_i]$ to the base $\mathfrak{g}$ is known for every $i \in \{2, \ldots, n\}$.

10. It implies that it is not known if $\{[\prod_{i=1}^{n} \mathfrak{I}_i^{e_i}] \mid (e_1, \ldots, e_n) \in [-B, B]^n\} = \mathcal{C}\ell(\mathcal{O})$ holds, if a uniform sampling in $[-B, B]^n$ determines a uniform distribution in $\mathcal{C}\ell(\mathcal{O})$, and, a priori, what are the collisions in $\{[\prod_{i=1}^{n} \mathfrak{I}_i^{e_i}] \mid (e_1, \ldots, e_n) \in [-B, B]^n\}$. The validity of the first two points is generally assumed on the base of heuristic arguments.

11. We observe that 3, 37, 1407181, 51593604295295867744293584889, 315994145046819958530082787455878322049 09 are primes, and that $\mathcal{C}\ell(\mathbb{Z}[\sqrt{-p}])$ is an abelian group.

Being $\mathcal{C}\ell\left(\mathbb{Z}\left[\sqrt{-p}\right]\right)$ isomorphic to the additive group $\mathbb{Z}_N$, where N is the class number of $\mathbb{Z}\left[\sqrt{-p}\right]$, each of its elements is identified with an integer in $\mathbb{Z}_N$. In particular, we denote by e the integer corresponding to the secret key $(e_1, \ldots, e_n)$, i.e. $\mathfrak{g}^e = \prod_{i=1}^n [\mathfrak{J}_i^{e_i}]$. As we discuss below, the existence of a canonical representation for elements of $\mathcal{C}\ell\left(\mathbb{Z}\left[\sqrt{-p}\right]\right)$ is essential for the CSIDH-based digital signature CSI-FiSh [6].

CSI-FiSh is a Fiat-Shamir signature, i.e. it is obtained turning an isogeny-based interactive identification protocol, sketched by Stolbunov in his PhD thesis [39], into a non interactive one by means of the Fiat-Shamir transformation [20]. The two actors of the interactive protocol are a prover, producing a proof σ by means of their private key $(e_1, \ldots, e_n)$, and a verifier, who verifies the validity of the proof σ with respect to the prover's public key $[E_S] = [\prod_{i=1}^n \mathfrak{J}_i^{e_i}] \star [E_0]$.

According to Stolbunov's proposal, while producing the proof σ, the verifier samples a random ephemeral private key $(r_1, \ldots, r_n)$ and computes the commitment $[E_{\mathsf{com}}] = [\prod_{i=1}^n \mathfrak{J}_i^{r_i}] \star [E_0]$, which will be part of the proof. Furthermore, depending on the value of a flag bit, the proof will contain also $(r_1, \ldots, r_n)$ or $(e_1 - r_1, \ldots, e_n - r_n)$. To verify the validity of σ, the verifier checks that $[\prod_{i=1}^n \mathfrak{J}_i^{r_i}] \star [E_0]$ is equal to $[E_{\mathsf{com}}]$ (part of the proof), or that $[\prod_{i=1}^n \mathfrak{J}_i^{e_i - r_i}] \star [E_{\mathsf{com}}]$ is equal to $[E_S]$.

Unfortunately, the simple scheme sketched above is flawed, as the vector $(e_1 - r_1, \ldots, e_n - r_n)$ may leak information about the private key $(e_1, \ldots, e_n)$. In order to fix this issue, an initial remedy was proposed by De Feo and Galbraith in [15]. It consists into adopting a redundant representation of class group elements and performing rejection sampling. The result is a compact scheme for which, however, proof generation and verification are quite inefficient. On the other hand, the class number computation for the CSIDH-512 set of parameters allowed Beullens *et al.* to produce a much better fix.

In particular, $(e_1 - r_1, \ldots, e_n - r_n)$ is replaced by its canonical representative in $\mathbb{Z}_N$, that we will denote by rsp. The reception of rsp, however, constitutes a problem for the verifier, since they need to compute the action $[\mathfrak{g}^{\mathsf{rsp}}] \star [E_{\mathsf{com}}]$ which, in general, has exponential complexity. In order to obtain an equivalent representation $(f_1, \ldots, f_n)$ of rsp in $[-B, B]^n$ or in a slightly bigger set $[-B', B']^n$

(i.e. $[\mathfrak{g}^{\mathsf{rsp}}] = [\prod_{i=1}^{n} \mathfrak{I}_i^{f_i}]$), a lattice-based solution is applied.

The resulting signature scheme, CSI-FiSh, enjoys practical efficiency in both signature generation and verification, while maintaining the short signature size offered by SeaSign. Recently, a CSI-FiSh variant named Lossy CSI-FiSh was proposed by El Kaafarani *et al.* [18]. Lossy CSI-FiSh provides a better security proof while maintaining almost the same efficiency of CSI-FiSh. The name of this protocol is due to the fact that it relies on a *lossy* variant [12] of the identification protocol discussed above.

6.1. Efficient *smooth* representation of $\mathfrak{g}^{\mathsf{rsp}}$

As already mentioned, the practical efficiency of CSI-FiSh is granted by the possibility to represent $[\mathfrak{g}^{\mathsf{rsp}}]$ as $[\mathfrak{I}_1^{f_1} \cdot \ldots \cdot \mathfrak{I}_n^{f_n}]$, for some $(f_1, \ldots, f_n) \in [-B', B']^n$, in an efficient way. Since CSI-FiSh is specific for the CSIDH-512 set of parameters, in the following we will use the concrete values 74 and 5 for n and B, respectively.

We observe that the cost for computing the action corresponding to a vector $(f_1, \ldots, f_{74}) \in \mathbb{Z}^{74}$ highly depends on the number of atomic isogenies to compute, i.e. on its L_1-norm $\sum_{i=1}^{74} |f_i|$. Hence, in order to find a representative vector for rsp that leads to an efficient class group computation, a strategy (possibly not the optimal one) is that of computing the closest vector to $(\mathsf{rsp}, 0, \ldots, 0)$ in the lattice

$$\Lambda := \left\{ (z_1, \ldots, z_{74}) \in \mathbb{Z}^{74} \middle| [\mathfrak{I}_1^{z_1} \cdot \mathfrak{I}_{74}^{z_{74}}] = [(1)] \right\}$$

with respect to the L_1-norm. We note that the knowledge of the lattice Λ is a side-product of the class number computation executed in [6], for which a reduced basis was then computed by Beullens *et al.*

A first approximation $(f'_1, \ldots, f'_{74})$ of the vector closest to $(\mathsf{rsp}, 0, \ldots, 0)$ can be obtained by applying the Babai's Nearest Plane algorithm [2] on the reduced basis for Λ. To further lower its L_1-norm, in [6] it is suggested the use of a second algorithm (see [17, 25]), detailed in Algorithm 4, which takes as input the vector $(f'_1, \ldots, f'_{74})$ together with a list of 10000 short vectors of Λ.

12. In a lossy identification protocol, the public-key space contains *lossy* keys, i.e. keys that do not admit a corresponding private key.

Algorithm 4 Randomized slicer for solving the closest vector problem

Input A list $\mathcal{S}$ of short vectors of Λ, $\underline{f}' = (f'_1, \ldots, f'_{74}) \in \mathbb{Z}^{74}$, the number M of maximum iterations.
Output Approximate closest lattice vector z to $\underline{f}'$.

Initialize $z \leftarrow (0, \ldots, 0)$
for $i = 0, \ldots, M-1$ do
    Sum a random vector in $\mathcal{S}$ to $\underline{f}'$ and obtain $\underline{f}''$.
    for $s \in \mathcal{S}$ do
        if $||\underline{f}'' - s||_1 < ||\underline{f}''||_1$ then
            $\underline{f}'' \leftarrow \underline{f}'' - s$
        end if
    end for
    if $||\underline{f}''||_1 < ||\underline{f}' - z||_1$ then
        $z \leftarrow \underline{f}' - \underline{f}''$
    end if
end for
return z

We executed some computational experiments[13] to shed light on how the number of iterations M in the algorithm above affects the output vector and the cost for computing the corresponding group action[14]. In particular, we sampled 100 uniform values for a in the set $\{0, \ldots, N-1\}$. For each of the obtained vectors $(a, 0, \ldots, 0)$, we executed the Babai's Nearest Plane algorithm, then two iterations of Algorithm 4, and then another two. The results are summarised below:

— when only the Babai's Nearest Plane algorithm is executed, the average cost for computing the class group action corresponding to the output vector is 257952 field multiplications;
— two iterations of Algorithm 4 lowers the average cost to 232111 field multiplications;
— for four iterations, the average cost is 227470 field multiplications.

13. Perfomed using the mathematics sofware system SageMath (version 8.5) on an Asus ZenBook 13 UX331U machine with Intel Core i7-8550U CPU @ 4 GHz.

14. This cost is computed evaluating the cost functions of the operations required by the class group computation, for which we used Algorithm 1 enhanced with the improvements described in Section 4.1.

Hence, two iterations of Algorithm 4 saved 25841 field multiplications on average, while another two iterations only saved 4641 field multiplications on average, while roughly doubling the execution time[15].

For a smaller set of vectors $(a, 0 \ldots, 0)$, we computed the average cost for computing the class group actions corresponding to the vectors obtained after applying the Babai's Nearest Plane algorithm and both 10 and 100 iterations of Algorithm 4. In the first case (10 iterations), the resulting average cost is 222037 field multiplications, in the second case (100 iterations) it is 208240. Given that the number of saved field multiplications is not massive, trading a smaller L_1-norm for 10 or 100 costly iterations of Algorithm 4 seems not convenient.

As conclusive remarks, we report that the attempt of replacing the L_1 norm with a different one (L_∞ or $L_1 + L_\infty$) did not provide any significant improvement. Moreover, we obtained roughly the same distribution of positive and negative exponents in all the close vectors we obtained, independently of the considered norm.

7. Conclusions

The isogeny-based protocol CSIDH exploits the action of the ideal class group of a quadratic order on a set of supersingular elliptic curves for the exchange of cryptographic keys. CSIDH enjoys short keys, requires small bandwidth and its running time is in the realm of practicality. However, the scheme is still far from being competitive with other post-quantum cryptosystems (for example those based on lattices) in terms of efficiency. The main reason for this gap is the high computational cost of the class group action.

Since the proposal of the CSIDH scheme in 2018, several papers focusing on improving the computation of the class group action and making it constant time have appeared. Furthermore, a record class group computation led to the first practical isogeny-based signature

15. We note that we did not precisely quantify the difference between the field multiplications saved and those necessary for the extra iterations. This computation is left for future research.

scheme. In this paper we reviewed the mathematical and algorithmic aspects of some of these contributions in an unified treatment.

The above mentioned advancements have slightly improved the efficiency of the original CSIDH implementation and that of its constant-time variants. However none of them have determined a significant break-trough, and this might be due to the fact the all have as backbone the original CSIDH implementation. We speculate that a major contribution would need a change of paradigm, likely coming from number-theoretic results. We hope that this work would contribute to triggering further research on the topic.

In Section 6 we discussed the procedure proposed in [6] to obtain a convenient representation of a vector of the form $(a, 0, \ldots, 0)$, providing some experimental results. The natural question which arises is whether the proposed reduction is optimal. This remains an open question that would benefit from a modeling of the dependence of the class action cost on the probability of finding rational points with suitable orders.

Titolo Volume
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000002
pag. 45–48 (xx 9999)

Bibliografia

[1] Azarderakhsh, R., Jao, D., Kalach, K., Koziel,B., and Leonardi, C., Key Compression for Isogeny-Based Cryptosystems, In Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography (AsiaPKC '16), 2016, Association for Computing Machinery, New York, NY, USA, 1–10.

[2] Babai, L., On Lovász' Lattice Reduction and the Nearest Lattice Point Problem, Combinatorica, 6(1), 1–13, 1986.

[3] Bernstein, D. J., De Feo, L., Leroux, A., Smith, B., Faster computation of isogenies of large prime degree, In Fourteenth Algorithmic Number Theory Symposium (*to appear*), 2020.

[4] Bernstein, D. J., Hamburg, M., Krasnova, A., Lange, T., Elligator: Elliptic-curve points indistinguishable from uniform random strings, In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 967–980, 2013.

[5] Bernstein, D. J., Lange, T., Martindale, C., Panny, L., Quantum circuits for the CSIDH: optimizing quantum evaluation of isogenies, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019, 409–441. Springer, Cham.

[6] Beullens, W., Kleinjung, T., Vercauteren, F., CSI-FiSh: Efficient Isogeny based Signatures through Class Group Computations, In International Conference on the Theory and Application of Cryptology and Information Security, 2019, 227–247. Springer, Cham.

[7] Castryck, W., Decru, T., CSIDH on the surface, In International Conference on Post-Quantum Cryptography, 2020, 111–129. Springer, Cham.

[8] Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J., CSIDH: an efficient post-quantum commutative group action, In International Conference on the Theory and Application of Cryptology and Information Security, 2018, 395–427. Springer, Cham.

[9] Cervantes-Vázquez, D., Chenu, M., Chi-Domínguez, J. J., De Feo, L., Rodríguez-Henríquez, F., Smith, B., Stronger and faster side-channel protections for CSIDH, In International Conference on Cryptology and Information Security in Latin America, 2019, 173–193. Springer, Cham.

[10] Chi-Domínguez, J. J., Rodríguez-Henríquez, F., Optimal strategies for CSIDH, Cryptology ePrint Archive, Report 2020/417, https://eprint.iacr.org/2020/417, 2020.

[11] Childs, A., Jao, D., Soukharev, V., Constructing elliptic curve isogenies in quantum subexponential time, Journal of Mathematical Cryptology, 8(1), 1–29, 2014.

[12] Costello, C., Hisil, H., A simple and compact algorithm for SIDH with arbitrary degree isogenies, In International Conference on the Theory and Application of Cryptology and Information Security, 2017, 303–329. Springer, Cham.

[13] Couveignes, J. M., Hard Homogeneous Spaces, Cryptology ePrint Archive, Report 2006/291, https://eprint.iacr.org/2006/291, 2006.

[14] Cox, D. A., Primes of the form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication, John Wiley & Sons, Ltd, 1997.

[15] De Feo, L., Galbraith, S. D., SeaSign: Compact isogeny signatures from class group actions, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019, 759–789. Springer, Cham.

[16] Delfs, C., Galbraith, S. D., Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$, Designs, Codes and Cryptography, 78(2), 425–440, 2016.

[17] Doulgerakis, E., Laarhoven, T., de Weger, B., Finding closest lattice vectors using approximate Voronoi cells, In International Conference on Post-Quantum Cryptography, 2019, 3–22. Springer, Cham.

[18] El Kaafarani, A., Katsumata, S., Pintore, F., Lossy CSI-FiSh: Efficient Signature Scheme with Tight Reduction to Decisional CSIDH-512, In IACR International Conference on Public-Key Cryptography, 2020, 157–186. Springer, Cham.

[19] Enge, A., Elliptic Curves and Their Applications to Cryptography: An Introduction, Kluwer Academic Publishers, 1999.

[20] Fiat, A., Shamir, A., How To Prove Yourself: Practical Solutions to Identification and Signature Problems, In Conference on the theory and application of cryptographic techniques, 1986, 186–194. Springer, Berlin, Heidelberg.

[21] Hutchinson, A., LeGrow, J., Koziel, B., Azarderakhsh, R., Further optimizations of CSIDH: a systematic approach to efficient strategies, permutations, and bound vectors, In International Conference on Applied Cryptography and Network Security, 2020, 481–501. Springer, Cham.

[22] Jao, D., De Feo, L., Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies, In International Work-

shop on Post-Quantum Cryptography, 2011, 19–34. Springer, Berlin, Heidelberg.

[23] Kuperberg, G., A subexponential-time quantum algorithm for the dihedral hidden subgroup problem, SIAM J. Comput., 35(1):170–188, 2005.

[24] Kuperberg, G., Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem, TQC, 22:20–34, 2005.

[25] Laarhoven, T., Sieving for closest lattice vectors (with preprocessing), In International Conference on Selected Areas in Cryptography, 2016, 523–542. Springer, Cham.

[26] Liu, D., Song, T., Dai, Y., Isomorphism and Generation of Montgomery-Form Elliptic Curves Suitable for Cryptosystems, Tsinghua Science & Technology, vol. 10, no. 2, 145–151, 2005.

[27] Meyer, M., Campos, F., Reith, S., On lions and elligators: An efficient constant-time implementation of CSIDH, In International Conference on Post-Quantum Cryptography, 2019, 307–325. Springer, Cham.

[28] Meyer, M., Reith, S., A faster way to the CSIDH, In International Conference on Cryptology in India, 2018, 137–152. Springer, Cham.

[29] Milne, J. S., Algebraic Number Theory (v3.08), Available at www.jmilne.org/math/, 2020.

[30] Moody, D., Shumow, D., Analogues of Velu's Formulas for Isogenies on Alternate Models of Elliptic Curves, Mathematics of Computation, 85(300), 1929–1951, 2016.

[31] Nakagawa, K., Onuki, H., Takayasu, A., Takagi, T., L_1-Norm Ball for CSIDH: Optimal Strategy for Choosing the Secret Key Space, Cryptology ePrint Archive, Report 2020/181, https://eprint.iacr.org/2020/181, 2020.

[32] Onuki, H., Aikawa, Y., Yamazaki, T., Takagi, T., A Faster Constant-Time Algorithm of CSIDH Keeping Two Points, In International Workshop on Security, 2019, 23–33. Springer, Cham.

[33] Onuki, H., Takagi, T., On collisions related to an ideal class of order 3 in CSIDH, In International Workshop on Security, 2020, 131–148. Springer, Cham.

[34] Peikert, C., He gives C-sieves on the CSIDH, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020, 463–492. Springer, Cham.

[35] Rostovtsev, A., Stolbunov, A., Public-key cryptosystem based on isogenies, Cryptology ePrint Archive, Report 2006/145, https://eprint.iacr.org/2006/145, 2006.

[36] Schoof, R., Nonsingular Plane Cubic Curves over Finite Fields, J. Comb. Theory, Ser. A, 46(2), 183–211, 1987.

[37] Shor, P. W., Algorithms for Quantum Computation: Discrete Logarithms and Factoring, In Proceedings 35th annual symposium on foundations of computer science, 1994, 124–134. IEEE.

[38] Silverman, J., The Arithmetic of Elliptic Curves, (Vol. 106), 2009. Springer Science & Business Media.

[39] Stolbunov, A., Cryptographic Schemes Based on Isogenies, PhD thesis, Jan. 2012, doi:10.13140/RG.2.2.20826.44488.

[40] Tate, J., Endomorphisms of Abelian Varieties over Finite Fields, Inventiones mathematicae, vol. 2, 134–144, 1966.

[41] Washington, L. C., Elliptic Curves: Number Theory and Cryptography, Second Edition, Chapman and Hall/CRC, 2008.

[42] Waterhouse, W. C., Abelian varieties over finite fields, Annales scientifiques de l'École Normale Supérieure, vol. 2, no. 4, 521–560, 1969.